

## Claim Amendments

1 – 40. (cancelled).

41. (new) A file system for managing a storage subsystem's RAID functionality such that files stored in the storage subsystem may have different RAID levels of protection thereby optimizing utilization of storage subsystem's storage units, the storage subsystem comprising a plurality of storage units organized in an array, the file system comprising:

a policy manager configured with block-level rules for determining a RAID level of protection for data of a file to be written to the storage subsystem, wherein the storage system is configured to store data of files having different RAID levels on the same storage unit or set of storage units and the file system is configured as a write anywhere log structured filing system of the storage subsystem;

a workflow manager that determines the RAID level for how data of said file should be written to the storage units by querying the policy manager for rules relating to storing a new file having parameters associated with data of said file;

a RAID director that implements RAID functionality for said file (based on the determined RAID level) and determines to store data of said file: storage units to use, parity codes to be generated, and the block-level size of write operations to the storage subsystem;

a RAID engine that processes data of said file using said parity codes to generate RAID parity data based on the determined RAID level; and

an access manager that monitors types of access patterns being made to data of files stored in the storage subsystem and provides the policy manager with access pattern information for data of said stored files thereby allowing the policy manager to determine what RAID stripe sizes and write coalescing would improve the storage subsystem's performance.

42. (new) The file system of claim 41, wherein in response to a file system request to create a file, the workflow manager queries the policy manager to determine whether the file to be created should be protected by a specific RAID level and whether it should be coalesced with other commands before data of the file to be created is written to the storage subsystem.

43. (new) The file system of claim 42, wherein in response to a file system request to write data to a file already created, the RAID manager determines if write data should be updated in place or relocated based on the created file's RAID level.

44. (new) The file system of claim 41, wherein the RAID director adjusts the RAID characteristics of a file storing a log of said write anywhere log structured filing system, said adjusting to optimize the storage subsystem's performance.

45. (new) The file system of claim 44, wherein said RAID characteristics are selected from the group consisting of: unprotected, mirrored, mirrored N ways, RAID level, and RAID stripe size.

46. (new) The file system of claim 41, further comprising:

a space manager that maintains storage information for blocks in the storage units, and determines block locations to store data of said file or to update a file stored in the storage subsystem;

47. (new) The file system of claim 41, further comprising:

a metadata manager that manages generation and updating of inodes contained in a file, the inodes describe information of files stored in the storage subsystem, the information of files comprising: a RAID level applied to a file, a stripe size of data of a file, block locations for data of a stored, and block locations of parity data for a file.

48. (new) The file system of claim 41, further comprising:

a system interface that receives I/O requests from a host system for the storage subsystem.

49. (new) The file system of claim 48, further comprising:

a first write cache that buffers data received by the system interface for a file to be stored in a storage system.

50. (new) The file system of claim 49, further comprising:

a second write cache that buffers the processed data of said file and said information generated by the RAID engine;

51. (new) The file system of claim 41, further comprising:

a device interface that issues file read and/or write storage commands to the storage units.

52. (new) The file system of claim 41, wherein the block-level rules are based on a size of a file to be stored by the file system.

53. (new) The file system of claim 41, wherein the block-level rules are based on contents of data of a file to be stored by the file system.

54. (new) The file system of claim 44, further comprising:

a cleaner daemon that reorders data if sections of said file storing said log become fragmented.

55. (new) The file system of claim 51, where the cleaner daemon cleans up partially used RAID stripes, if at least two discrete files are stored in a single RAID stripe and after at least one of the discrete files is deleted such that the RAID stripe is only part full and can be coalesced with data from another RAID stripe and rewritten, thereby creating additional available storage in the storage subsystem.

56. (new) The file system of claim 41, wherein the RAID level is selected from the group consisting of: RAID 0, RAID 1, RAID 5, and RAID 6.